1. Tool Use

Introduction:

Key points about use of the standard modeling tool are:

- All-Fusion ERwin Data Modeler is used on individual workstations.
- Agency standard ERwin data model templates (.er1 files) are required starting points for new Logical and Physical models. These templates include the standard UDPs which are required components of all new models.
- Built-in features of All-Fusion ERwin Data Modeler include extensive documentation and help facilities. More tips are available at the vendor's Internet website.
- All-Fusion ERwin Data Modeler is software designed for use by professional data analysts. An appropriate level of product experience plus understanding of data architecture principles and methods are required to produce efficient and effective data model products.
- IDEF1X notation is the required standard to be used within CMS.
- Data Administration provides modeling tool software licenses and registration information.

Contact Data Administration for assistance with activities and standard data modeling and modeling software tools.

NOTE: There are references within this section that refer the reader to the Operating Procedures and Guidelines section. Please download the Operating Procedures and Guidelines section to view these references.

The topics for data modeling tool use are:

1.1. Data Modeling Tool Standard for using User Defined Properties

Introduction
The standard User Defined Properties (UDPs) offer a range of improved information for and about the logical and physical data models. At the minimum the UDPs should provide:

1) Better linkage between attributes and project business requirements
2) A clearer connection between attributes/columns and their primary data source
3) Specification of an entity's security classification
4) Improved linkage between logical model entities and physical model tables.

The application of the UDPs can easily be broken into two broad categories:

- New systems being developed
- Legacy systems already in production that are being revised.

New Systems Being Developed
After October 1, 2005, any new application being developed for operation within CMS's data center is encouraged to incorporate the standard UDPs in both the logical and physical data models. (Refer to DM G-020 Guideline for Using ERwin Complete Compare to Create a Brand New Data Model.)

Legacy Systems
Applications in operation prior to October 1, 2005 are not expected to retroactively incorporate the standard UDPs. However, as these applications are revised the new releases should incorporate the UDPs into the data models whenever any one of the following criteria are met:

- A major application release occurs. Major release is defined as a revision that causes the application's version number to increase as follows.
  Example: Application version 1.2 becomes Application version 2.0

- The application revision causes a new subject area to be added to the application's data models

- The application revision causes the addition of three or more new entities/tables to one or more subject areas in the application's data models

Use of the standard UDPs only applies to the new data objects (entities, attributes, tables columns, etc.) being placed in the legacy applications' data models. Modelers are encouraged to supply the model level UDPs as well when significant maintenance is being performed.

The templates will often be used as a means of adding the standard UDPs into the numerous data models that are already in existence in various projects and systems. Considering this let us anticipate some possible concerns that may exist:

Concern #1: Suppose a model that pre-dates the new standard UDPs already has some UDPs of its own?

Answer to Concern #1: There should be no problem. The new standard UDPs will simply be imported so that they exist alongside the UDPs already in place in the model. The pre-existing UDPs will not be overlaid.

Concern #2: Suppose a model already has a UDP with a name *identical to* that of a standard UDP that is to be imported?

Answer to Concern #2: In that case the modeler will have the opportunity to deliberately select the identically named standard UDP and *not* import that UDP in the procedure described in DM G-021 Guideline for Using ERwin Complete Compare to Import Standard Logical Model UDPs.

Concern #3: Suppose a model already has a UDP that is differently named from a standard UDP, but identical in purpose? For instance, a logical model might have a "Business Need" UDP that stores the business requirements that have caused that attribute to be placed into the logical data model. This is the same purpose intended for the new attribute-level standard UDP named "Attribute Requirement ID".

Answer to Concern #3: As in "Concern #2" above, following the procedure described in DM G-021 Guideline for Using ERwin Complete Compare to Import Standard Logical Model UDPs, the modeler will deliberately avoid importing the same-purpose standard UDP. Then the modeler can change the name of the old UDP so that it has the same name as the new standard UDP. In this case, the old "Business Need" UDP would be renamed as "Attribute Requirement ID".

Viewing the UDPs in a Model

The UDPs become visible when you go to the ERwin toolbar and select "Model" and then "UDP Dictionary" from the menu below. In the subsequent popup screen you will immediately be shown the standard set of UDPs, the ones for the overall data "Model". In order to see the other UDPs, go to the dropdown list next to the "Class" area and select another class type (table, entity, attribute, column, etc.)

1.2. Data Modeling Tool Standard for Creating Conceptual Data Models

Introduction
All-Fusion ERwin Data Modeler is the standard data-modeling tool at CMS. Use of any other software tool for the purpose of developing data models is prohibited. This section describes creation of a project's *Data Models*, which has the purpose of showing the "big picture" perspective of project entities.

Responsibilities
The *Project Data Analyst* creates the *Conceptual Data Model*.

Relationships
Type *Verb Phrases* in lower case

Display Level
Display the model at the *Entity* level i.e. only entities and their relationships are to be shown on the Conceptual Data Model diagram.

Data Model Template File
The standard ERwin logical data model template file is Std_LDM_UDP_Template.er1. For databases in production or development before October 1, 2005, use the old standard naming convention file old_CMSGloss.nsm. For all databases being developed after October 1, 2005 use the new standard naming convention file CMS_Standard_Terms.nsm. These files are available from the DA Standard Tools page, which is accessible from the main Data Administration web page. The template file contains the CMS standard User Defined Properties (UDPs) format. The standard naming convention files enable the automatic naming features in the ERwin tool.

Refer to the *HELP* in the data modeling tool for information on how to implement and use the naming file.

The use of this model template is required for *Project Conceptual* or *Logical Data Models* being developed on or after October 1, 2005, which supports all new development projects. Models that will be used to modify databases that existed before October 1, 2005 are required to preserve any UDPs already defined in those previous models. Database enhancement projects are encouraged to incorporate and populate the new standard UDPs to the greatest practical extent. The new UDPs themselves can be added by using the ERwin compare utility to update the project LDM. (Refer to DM G-021 Guideline for Using ERwin Complete Compare to Import Standard Logical Model UDPs.) Then they can be populated using the UDP tab of the regular entity, attribute and model property update dialogues.

When you open either template up in ERwin all you will see is a blank screen. These templates contain absolutely no entitles, tables, or any other diagram objects. The templates exist solely to offer a set of standard UDPs that can be used in any data model.

The ERwin model properties for a Conceptual Data Model are to be specified according to the table which follows.

ERwin Model Properties for a Conceptual Data Model

Model Property	Format / Description	Reqd	UDP
LDM Business Contact Name	Enter the name of the organization and person who is responsible for approving the definitions in the model into the model UDP LDM Business Contact Name. (Graphic)	●	●
LDM Create Date	Enter the date designated by Central DA for the original model into the model UDP LDM Create Date. (This date will determine the standards that apply to the model.) (Graphic)	●	●
LDM DA Signoff Date	Enter the date when the LDM received its baseline/ most recent Central DA sign-off into the model UDP LDM Approval Date. (Graphic)	●	●
LDM Last Change Description	A brief narrative summarizing the nature of the changes resulting in the current model version. (Graphic)	●	●
LDM Modeler Contact Name	Enter the name of the organization and person who is responsible for developing the model into the model UDP Modeler Contact Name. (Graphic)	●	●
Logical Notation	Select IDEF1X. (Graphic)	●	
Model Definition	Provide a brief description of the business project whose high-level data requirements are represented by the entities and relationships to be diagramed in the Conceptual Data Model. The model definition describes the purpose and status of the model in a few sentences of text. Example: **Conceptual data model for the initial phase of the XYZ system.** (Graphic) See: DM OP- 028 Operating Procedure for Naming and Defining Data Models.	●	
Model History Options	Select all of these options. (Graphic)	●	
Model Name	A Conceptual Data Model is to be named in the following manner: *system acronym + ("relational"/"dimensional") + model type (EDM/CDM/LDM/PDM) + approval date (or the save date for models in development) in yyyymmdd format*. Example: **XYZ relational CDM 20040726**. (Graphic) See: DM OP- 028 Operating Procedure for Naming and Defining Data Models.	●	
Model Type	Designate the model as a pure logical or pure physical model, according to the kind of objects contained in the model. For a Conceptual Data Model, the type must be logical. (Graphic)	●	

Business Entities

The ERwin properties for each business entity in a Conceptual Data Model are to be specified according to the table which follows.

Entity Property	Format / Description	Reqd	UDP
Entity Business Contact Name	Enter the name of the organization and optionally the person who is responsible for approving the definitions of the Entity and its contained Attributes. (Graphic)	●	●
Entity Definition	The narrative explanation of the meaning of an instance of the Entity. Example: **Service Provider - A business licensed to dispense prescription drugs.** (Graphic) See: DM OP-008008 Operating Procedure for Defining Data Entities	●	
Entity Logical Only Switch	The indication of whether or not the Entity has a corresponding Table in a PDM. For Conceptual Data Models, set this switch to 'N' for each business entity. (Graphic)	●	
Entity Name	The user assigned symbolic identifier of the Entity. Type *Entity Names* in title case (the first letter of each term is in uppercase, the remaining letters in the term are in lowercase) throughout the model. (Graphic)	●	
Entity Requirement ID	A reference to the requirement(s) that justify the existence of the Entity in the model. The format of the reference to the DOORS Tracking ID is BR-#### for business requirements, SR-#### for system requirements and CR-#### for a change request. Multiple requirements are separated by semicolons. (Graphic)	●	●
Entity Security Category Description	A reference to the FISMA category scheme which describes the risk of unauthorized access, unauthorized modification or unavailability of the data represented by the Entity. The format of this UDP contains 3 values, separated by semicolons. E.g., CONFIDENTIALITY= impact; INTEGRITY= impact; AVAILABILITY= impact. Where impact has a value from the list: Low, Moderate, High, NA. Refer to http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf. (Graphic) See: DM OP-021 Operating Procedure for Assigning Information Security Categories	●	●
Physical Table Name	The name of the corresponding Table specified in the corresponding PDM for this LDM. If the physical model has not been created, this is the Table name as formed in accordance with the CMS standard naming conventions for the applicable DBMS. Not required for Conceptual Logical Data Models. Not required for "logical only" entities. (Graphic)	●	●

ERwin Screen Snapshots for Creating Conceptual Data Models

Exhibit 1. Model Properties for a Conceptual Logical Data Model

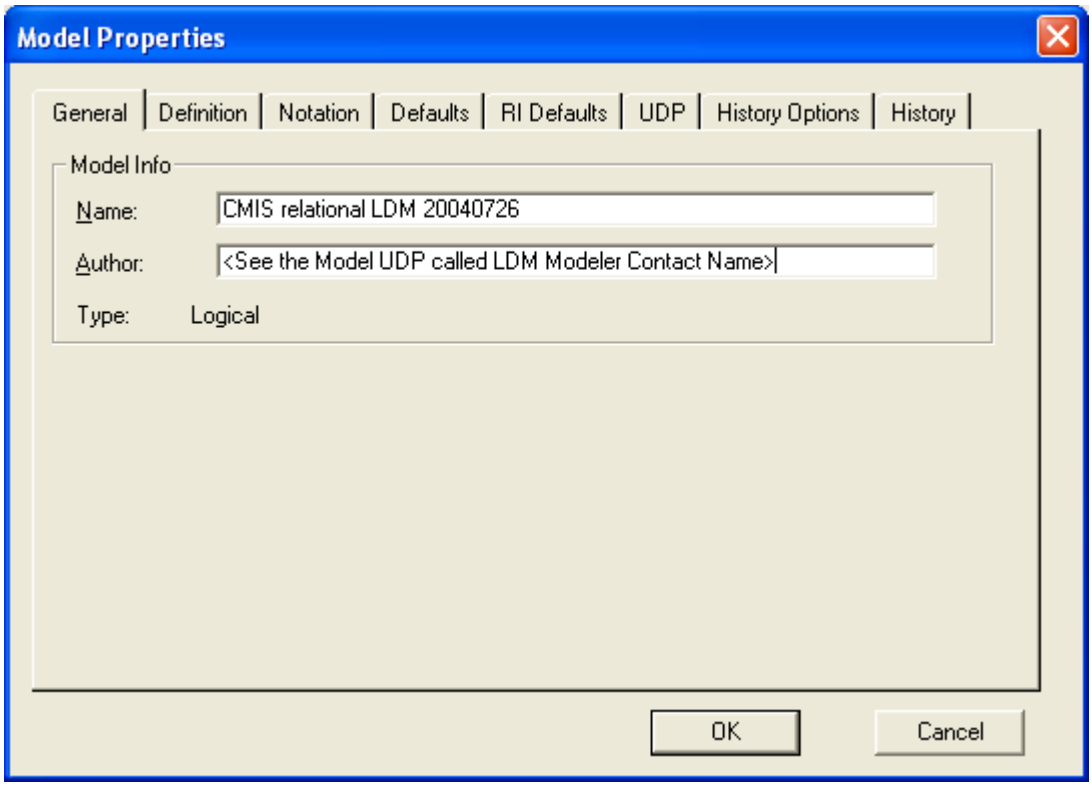


Exhibit 2. Model PropertiesProperties for a Conceptual Logical Data Model

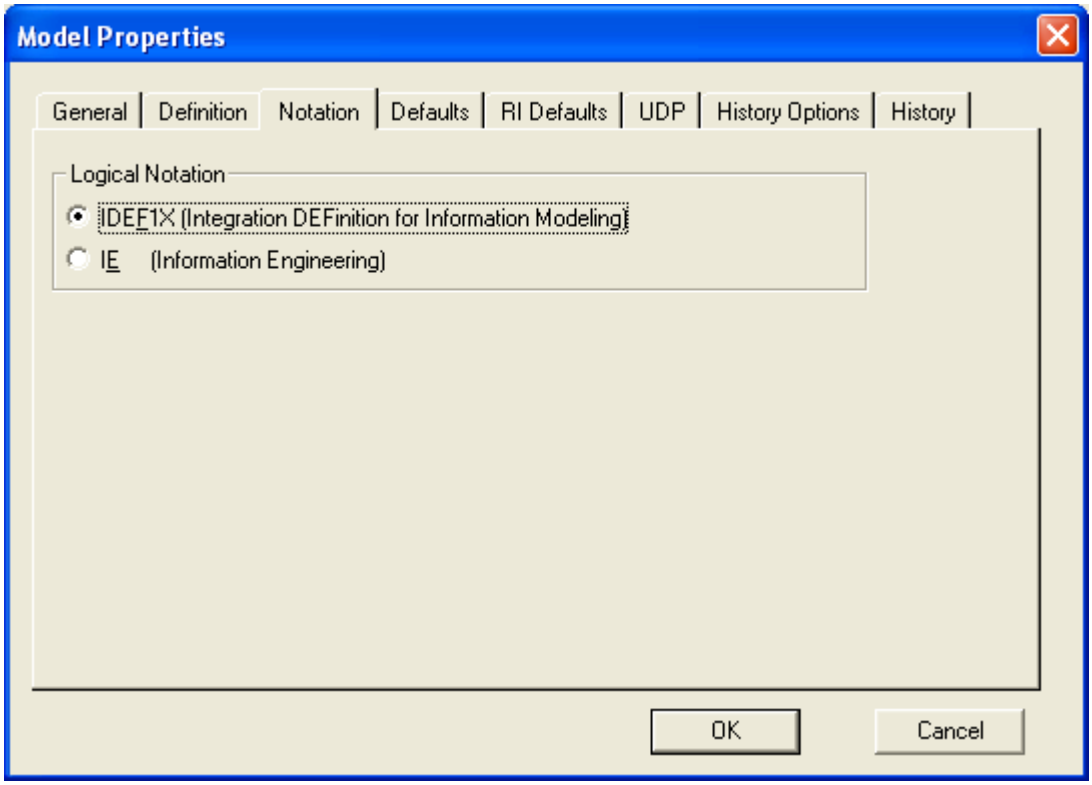


Exhibit 3. Type, Template Selection for aa Logical Data Model

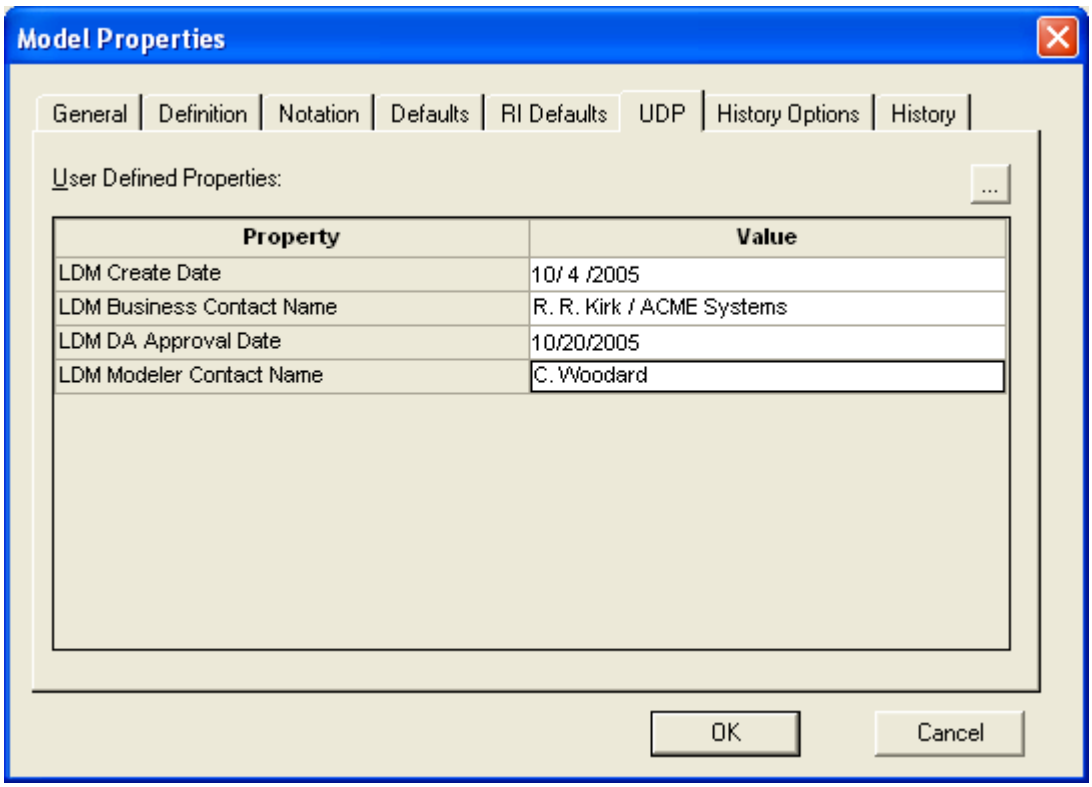


Exhibit 4. NotationNotation for aLogical Data Model

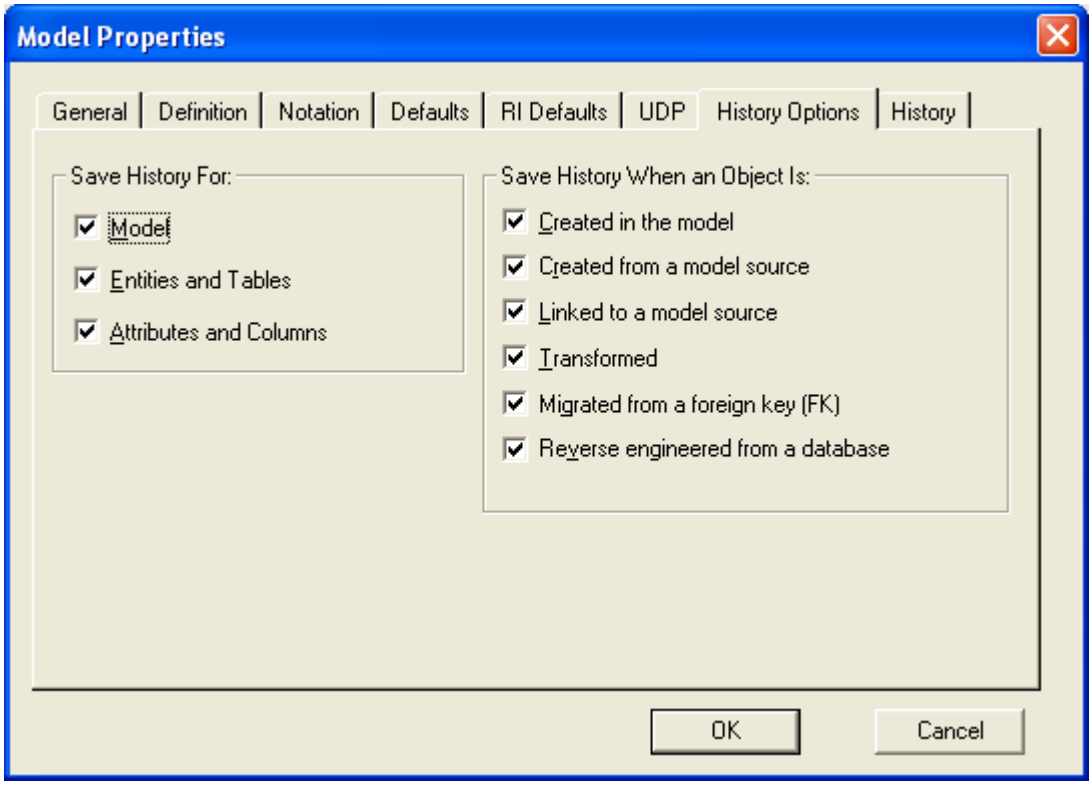


Exhibit 5. Model UDPs for aLogical Data Model

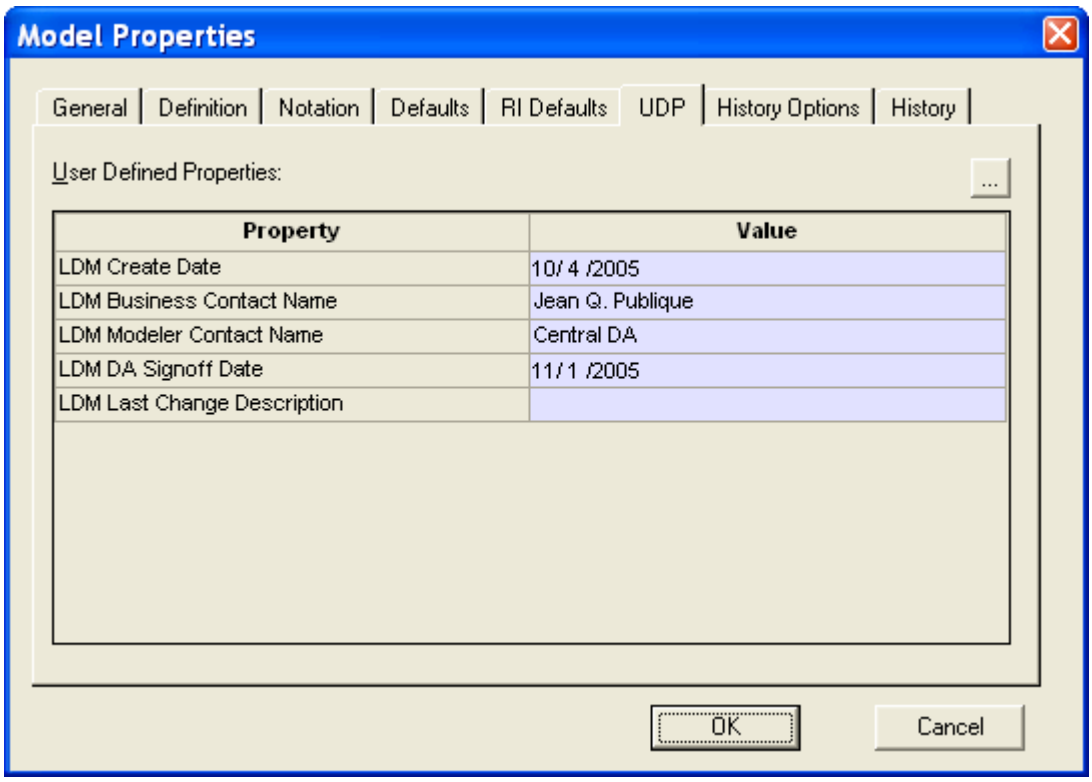


Exhibit 6. Model History Options for a Logical or Physical Data Model

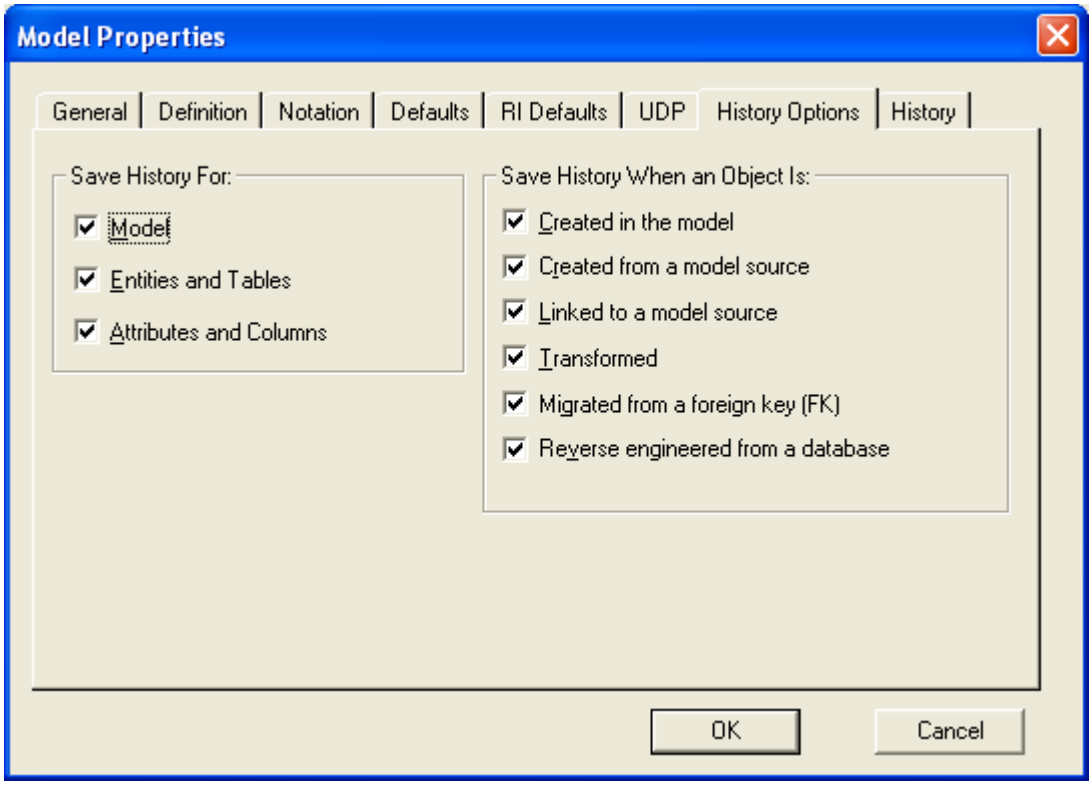


Exhibit 7. Example Entity and Relationship Names for a Conceptual Logical Data Model

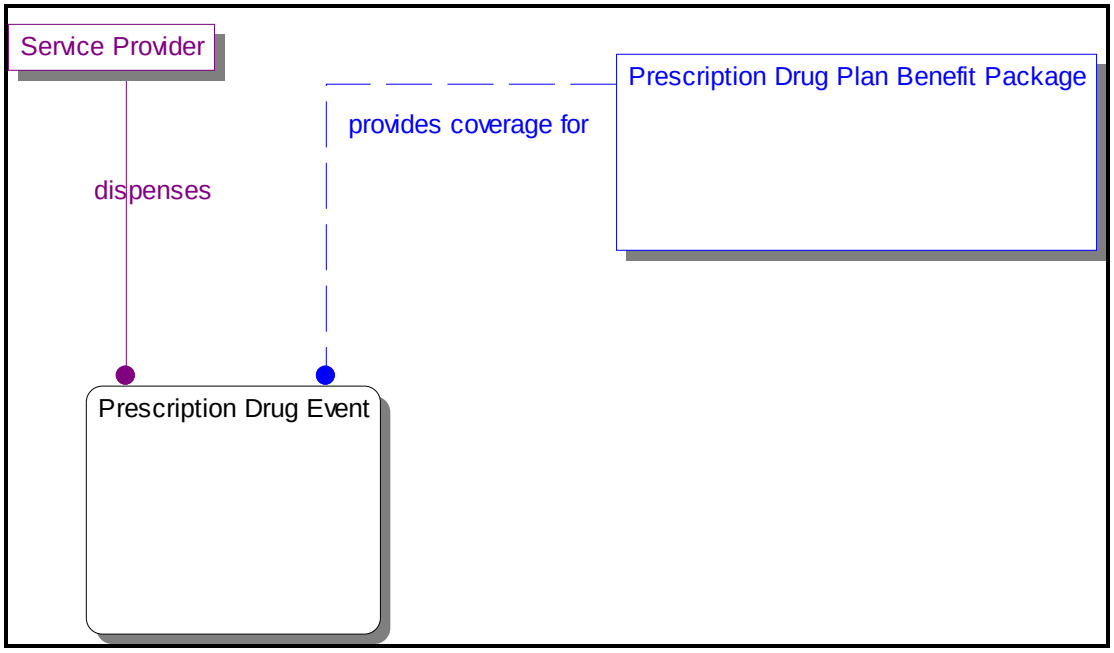


Exhibit 8. Entity Definition for a Logical Data Model

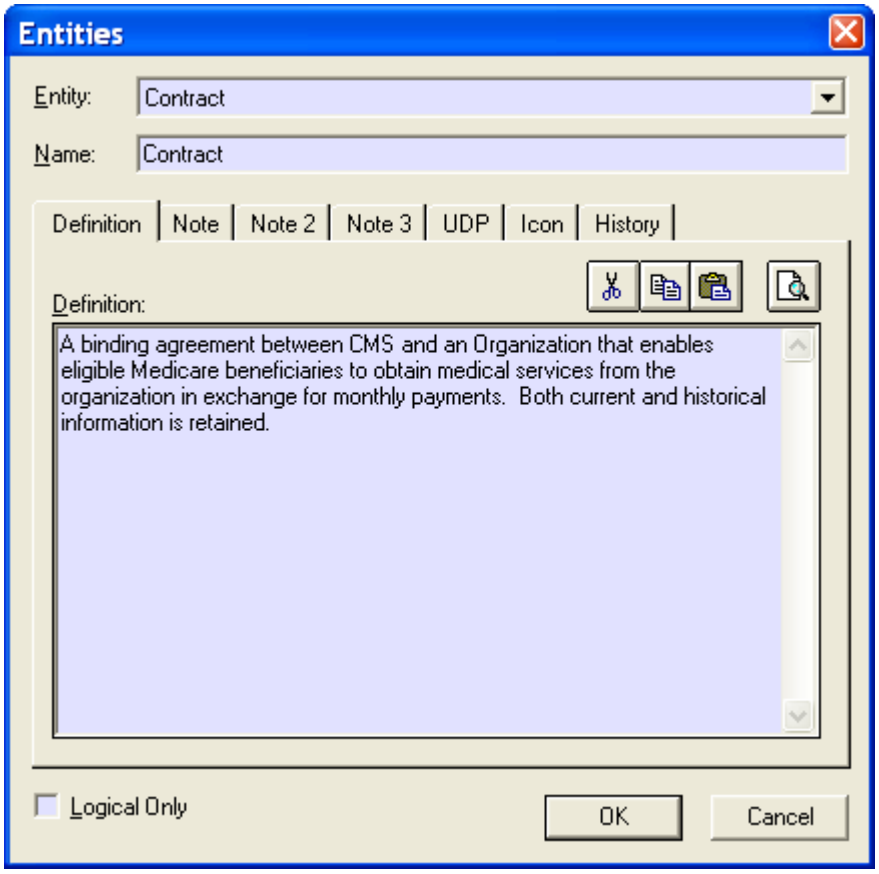
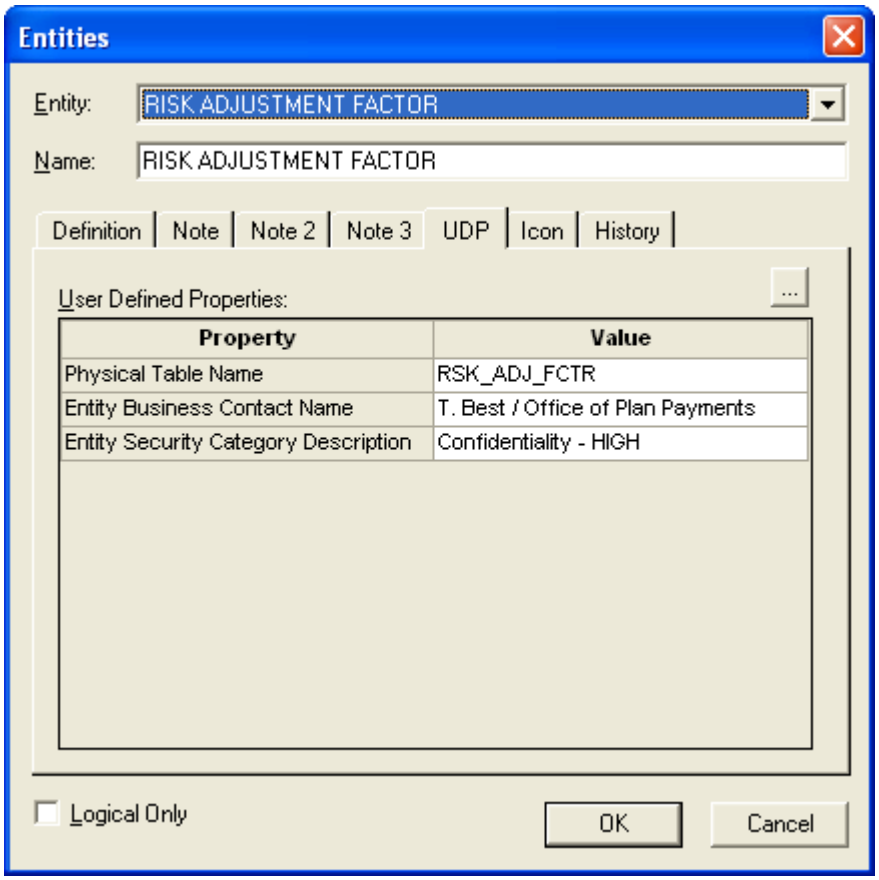


Exhibit 9. Entity UDPs for a Logical Data Model



1.3. Data Modeling tool standard for Creating Project Logical Data Models

Introduction
All-Fusion ERwin Data Modeler is the standard data-modeling tool at CMS. Use of any other software tool for the purpose of developing Logical Data Models is prohibited.

Responsibilities
All entries in the data modeling tool to create the Project Logical Data Model are performed by the *Project Data Analyst.*

Data Model Template File
A Model Template File and a Naming Standard File are available from the DA Standard Tools page, which is accessible from the main Data Administration web page. The template file contains the CMS standard User Defined Properties (UDPs). The Naming Standards File enables automatic naming features in the data modeling tool.

Refer to the *HELP* in the data modeling tool for information on how to implement and use the naming file.

The use of this model template is required for *Project Conceptual* or *Logical Data Models* being developed on or after October 1, 2005, which supports all new development projects. Models that will be used to modify databases that existed before October 1, 2005 are required to preserve any UDPs already defined in those previous models. Database enhancement projects are encouraged to incorporate and populate the new standard UDPs to the greatest practical extent. The new UDPs themselves can be added by using the ERwin compare utility to update the project LDM. (Refer to DM G-021 Guideline for Using ERwin Complete Compare to Import Standard Logical Model UDPs.) Then they can be populated using the UDP tab of the regular entity, attribute and model property update dialogues.

When you open either template up in ERwin all you will see is a blank screen. These templates contain absolutely no entitles, tables, or any other diagram objects. The templates exist solely to offer a set of standard UDPs that can be used in any data model.

The ERwin model properties for a project Logical Data Model are to be specified according to the table which follows.

Erwin Model Properties for a Project Logical Data Model

Model Property	Format / Description	Reqd	UDP
LDM Business Contact Name	Enter the name of the organization and person who is responsible for approving the definitions in the model into the model UDP LDM Business Contact Name. (Graphic)	●	●
LDM Create Date	Enter the date designated by Central DA for the original model into the model UDP LDM Create Date. (This date will determine the standards that apply to the model.) (Graphic)	●	●
LDM DA Signoff Date	Enter the date when the LDM received its baseline/ most recent Central DA sign-off into the model UDP LDM Approval Date. (Graphic)	●	●
LDM Last Change Description	A brief narrative summarizing the nature of the changes resulting in the current model version. (Graphic)	●	●
LDM Modeler Contact Name	Enter the name of the organization and person who is responsible for developing the model into the model UDP Modeler Contact Name. (Graphic)	●	●
Logical Notation	Select IDEF1X. (Graphic)	●	
Model Definition	Provide a brief description of the business project whose detailed data requirements are represented by the entities, attributes and relationships to be diagramed in the project Logical Data Model. The model definition describes the purpose and status of the model in a few sentences of text. Example: **Current approved LDM.** (Graphic) See: DM OP- 028 Operating Procedure for Naming and Defining Data Models.	●	
Model History Options	Select all of these options. (Graphic)	●	
Model Name	A Logical Data Model is to be named in the following manner: *system acronym + ("relational"/"dimensional") + model type (EDM/CDM/LDM/PDM) + approval date (or the storage date for models in development) in yyyymmdd format.* Example: **MBD relational LDM 20050720.**. (Graphic) See: DM OP- 028 Operating Procedure for Naming and Defining Data Models.	●	
Model Type	Designate the model as a pure logical or pure physical model, according to the kind of objects contained in the model. ERwin version 3.5.2 only manipulates a third type of model, called logical/ physical. For a Project Logical Data Model, the type must be logical. (Graphic)	●	

Data Entity Properties

The ERwin properties for each data entity in a Project Logical Data Model are to be specified according to the table which follows.

Entity Property	Format / Description	Reqd	UDP
Entity Business Contact Name	Enter the name of the organization and optionally the person who is responsible for approving the definitions of the Entity and its contained Attributes. (Graphic)	●	●
Entity Definition	The narrative explanation of the meaning of an instance of the Entity. Example: **Service Provider - A business licensed to dispense prescription drugs.** (Graphic) See: DM OP-008008 Operating Procedure for Defining Data Entities	●	
Entity Logical Only Switch	The indication of whether or not the Entity has a corresponding Table in a PDM. (Graphic)	●	
Entity Name	The user assigned symbolic identifier of the Entity. Type *Entity Names* in title case (the first letter of each word is in uppercase; the remaining letters in the word are in lowercase) throughout the model. (Graphic)	●	
Entity Requirement ID	A reference to the requirement(s) that justify the existence of the Entity in the model. The format of the reference to the DOORS Tracking ID is BR-#### for business requirements, SR-#### for system requirements and CR-#### for a change request. Multiple requirements are separated by semicolons. (Graphic)		●
Entity Security Category Description	A reference to the FISMA category scheme which describes the risk of unauthorized access, unauthorized modification or unavailability of the data represented by the Entity. The format of this UDP contains 3 values, separated by semicolons. E.g., CONFIDENTIALITY= impact; INTEGRITY= impact; AVAILABILITY= impact. Where impact has a value from the list: Low, Moderate, High, NA. Refer to http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf. (Graphic) See: DM OP-021 Operating Procedure for Assigning Information Security Categories	●	●
Physical Table Name	The name of the corresponding Table specified in the corresponding PDM for this LDM. If the physical model has not been created, this is the Table name as formed in accordance with the CMS standard naming conventions for the applicable DBMS. Not required for “logical only” entities. (Graphic)	●	●

Data Attribute Properties

The ERwin properties for each data attribute in a Project Logical Data Model are to be specified according to the table which follows.

Attribute Property	Format / Description	Reqd	UDP
Attribute Alias Name	An alternate business name used to refer to the attribute. (Graphic)		●
Attribute Data Source Name	The CMS database, external data feed, manual data entry process or software process from which the attribute takes its value. The format for CMS sources is CMS.database.table.column. The format for external sources is org.specification.pubdate.record.recordtype.field. Recordtype is used where there are multiple record formats and may be omitted if there is only one format for the record specification. The format for software processes is application.programname or packagename.servicename. The format for manual data entry processes is application.formname.fieldname. If this level of detail is not available, indicate the source in terms of a user role, type of stakeholder, organization, business process, system, program, database, file or data exchange standard. (Graphic)	●	●
Attribute Definition	The narrative explanation of the meaning of an instance of the attribute. (Graphic) Example: **Service Provider - A business licensed to dispense prescription drugs.** ***See:*** DM OP-010 Operating Procedure for Defining Data Attributes	●	
Attribute Derivation Text	The narrative explanation of any non-trivial logic used to transform information from one or more other attributes into the value for this attribute. (Graphic)		●
Attribute Domain Name	The name of the domain which defines the datatype, default value and valid values of the attribute. (Graphic)	●	
Attribute Logical Only Switch	The indication of whether or not the Attribute has a corresponding Column in a PDM. (Graphic)	●	
Attribute Name	The user assigned symbolic identifier of the Entity. Type *Attribute Names* in title case (the first letter of each word is in uppercase, the remaining letters inin the word are in lowercase) throughout the model. (Graphic) If the attribute is serving as a foreign key, assign a role name where necessary.	●	

Attribute Property	Format / Description	Reqd	UDP
Attribute Overriding Datatype Name	The name of the datatype which overrides the one specified in the Domain which governs the Attribute. (Graphic) (Required if no Domain is specified.)		
Attribute Overriding Default Value Name	The name of the Default Value which overrides the one specified in the Domain which governs the Attribute. (Graphic)		
Attribute Overriding Validation Rule Name	The name of the Validation Rule which overrides the one specified in the Domain which governs the Attribute. (Graphic)		
Attribute Required Switch	An indication of whether or not the Attribute must assume a non-null value when an entity instance is created. (Graphic)	●	
Attribute Requirement ID	A reference to the requirement(s) or change request(s) that justify the Attribute in the model. The format of the reference to the DOORS Tracking ID is BR-#### for business requirements, SR-#### for system requirements and CR-#### for a change request. Multiple requirements are separated by semicolons. (Graphic)	●	●
Physical Column Name	The name of the corresponding column specified in the corresponding PDM for this LDM. The format of this reference is ModelName.VersionDate.TableName.ColumnName. If the physical model has not been created, this is the Column name as formed in accordance with the CMS standard naming conventions for the applicable DBMS. Not required for "logical only" attributes. (Graphic)	●	●
Primary Identifier Switch	An indication of whether or not the attribute is part of the entity's primary identifier. (Graphic)	●	

Data Relationships

Verb phrase in all lower case
Relationship type
Specify identifying or non-identifying relationship.
Logical Only Indicator: Specify *Logical Only* status, if applicable.
Specify cardinality.

ERwin Screen Snapshots for Creating Logical Data Models

Exhibit 10. Model Properties for a Project Logical Data Model

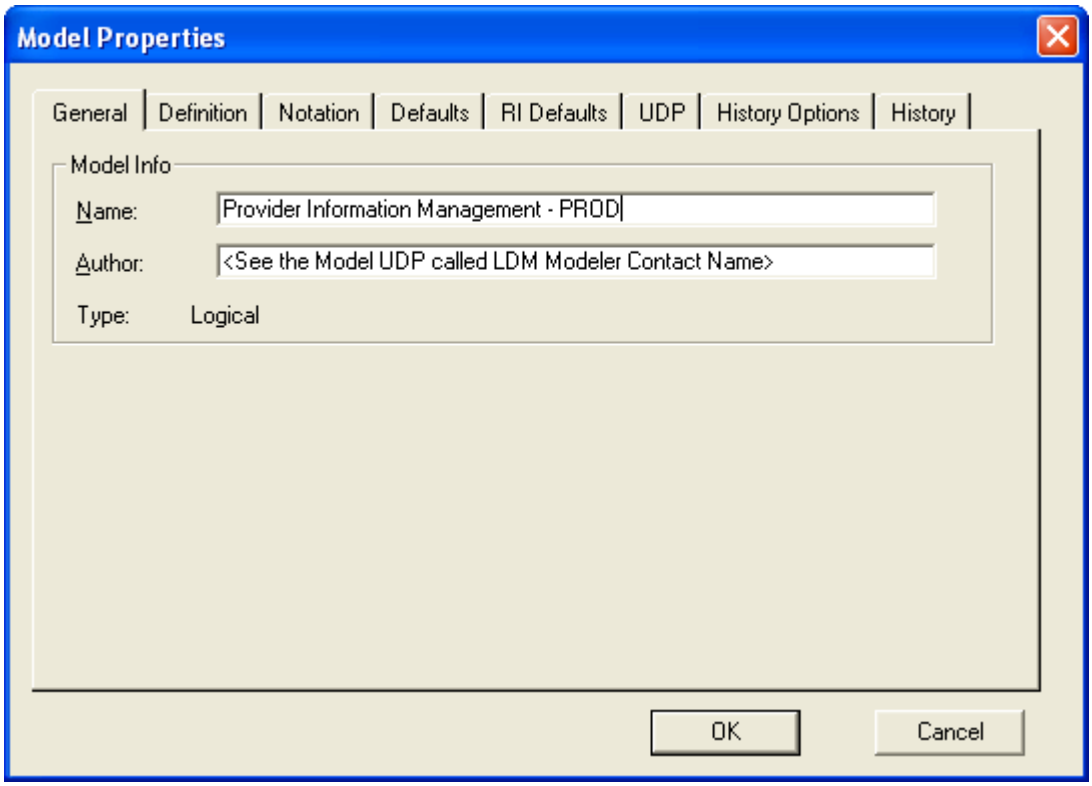


Exhibit 11. Example Entity and Relationship Names for a Project Logical Data Model

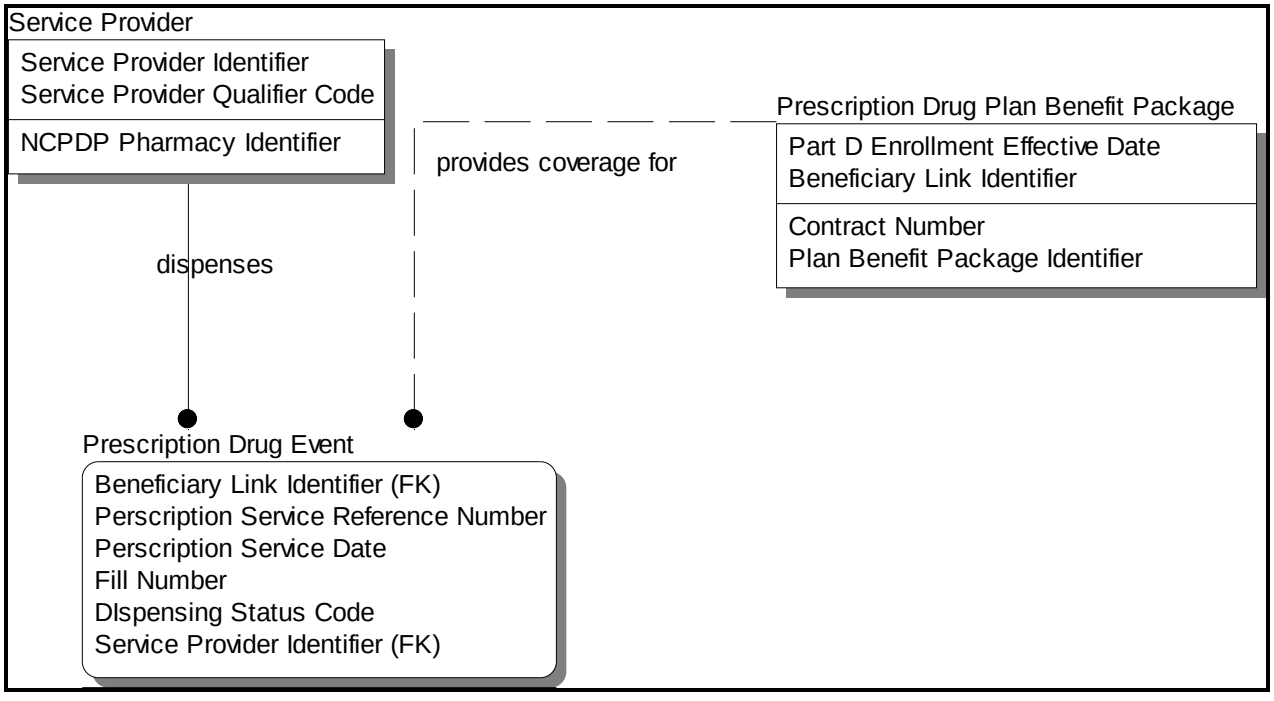


Exhibit 12. Example Attribute Definition for a Project Logical Data Model

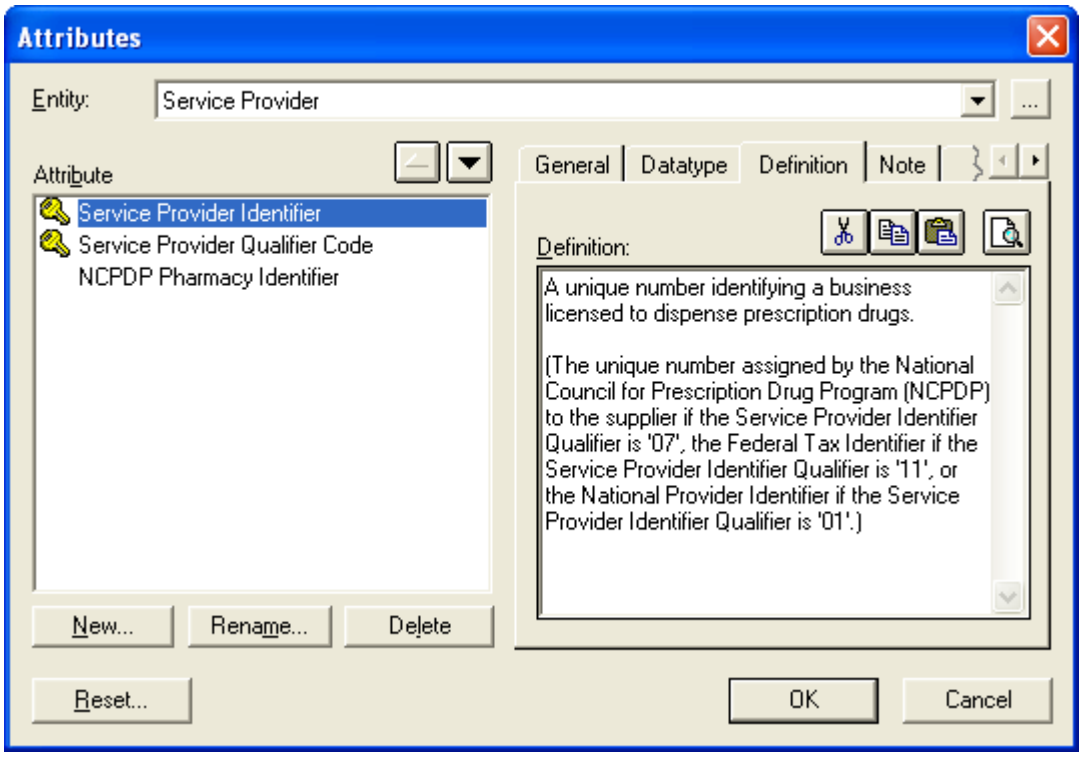


Exhibit 13. Attribute Domain Override for a Project Logical Data Model

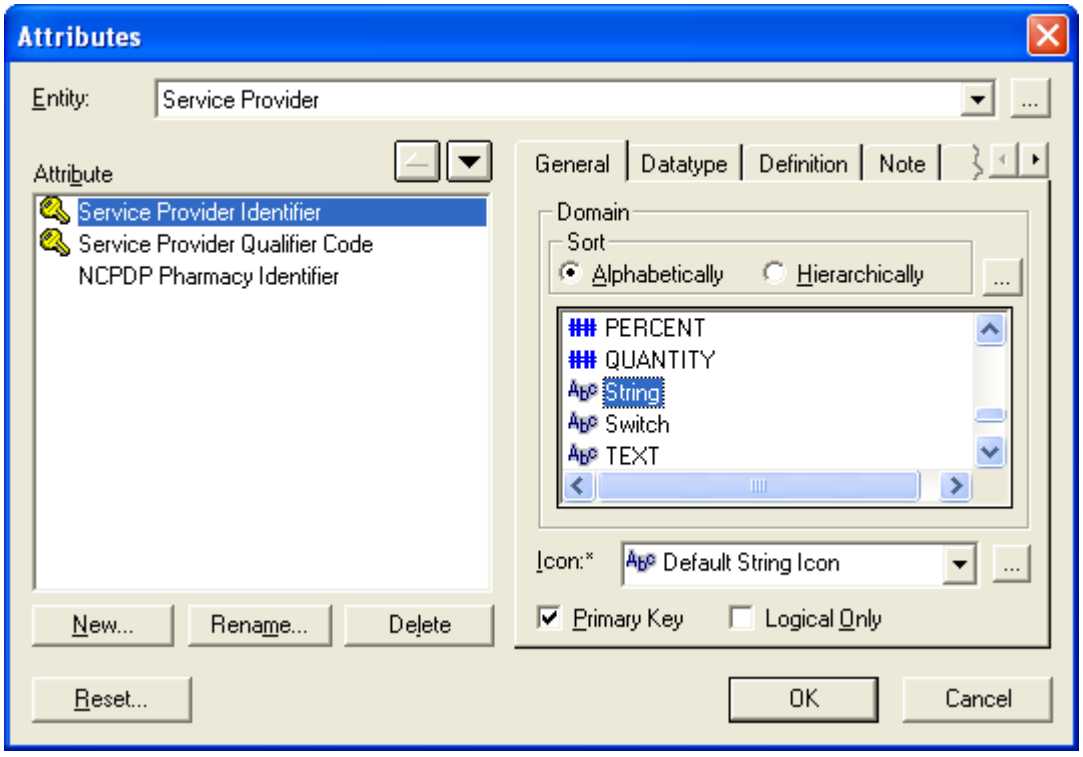


Exhibit 14. Attribute Validation Rule Override for a Project Logical Data Model

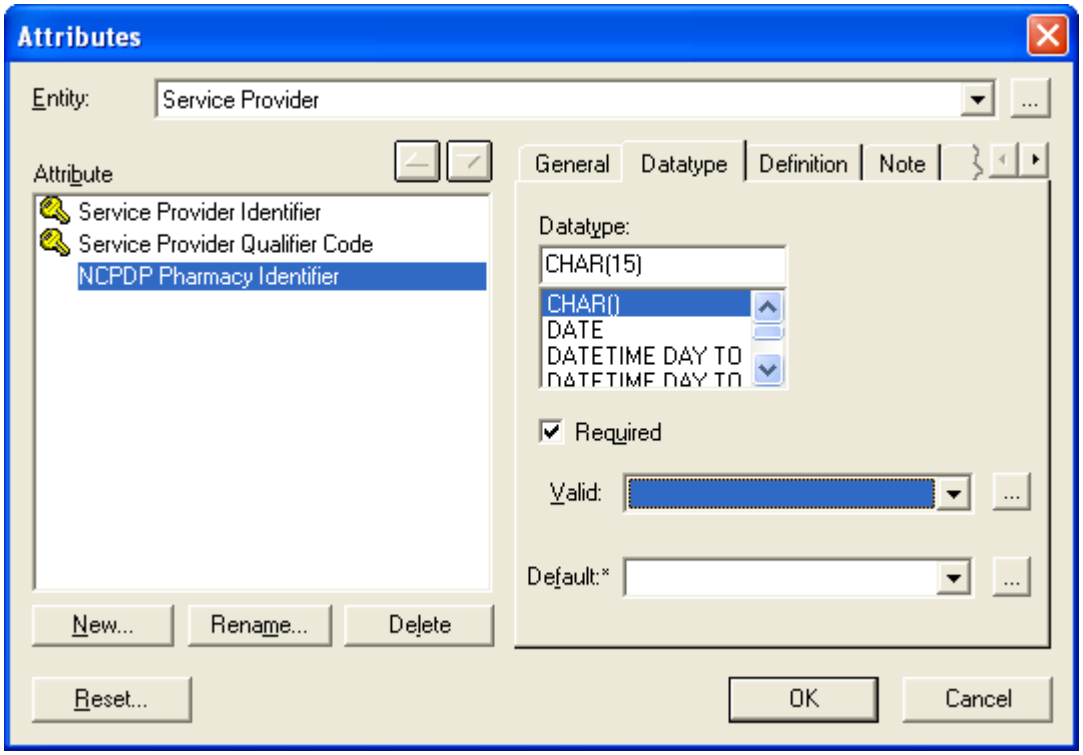
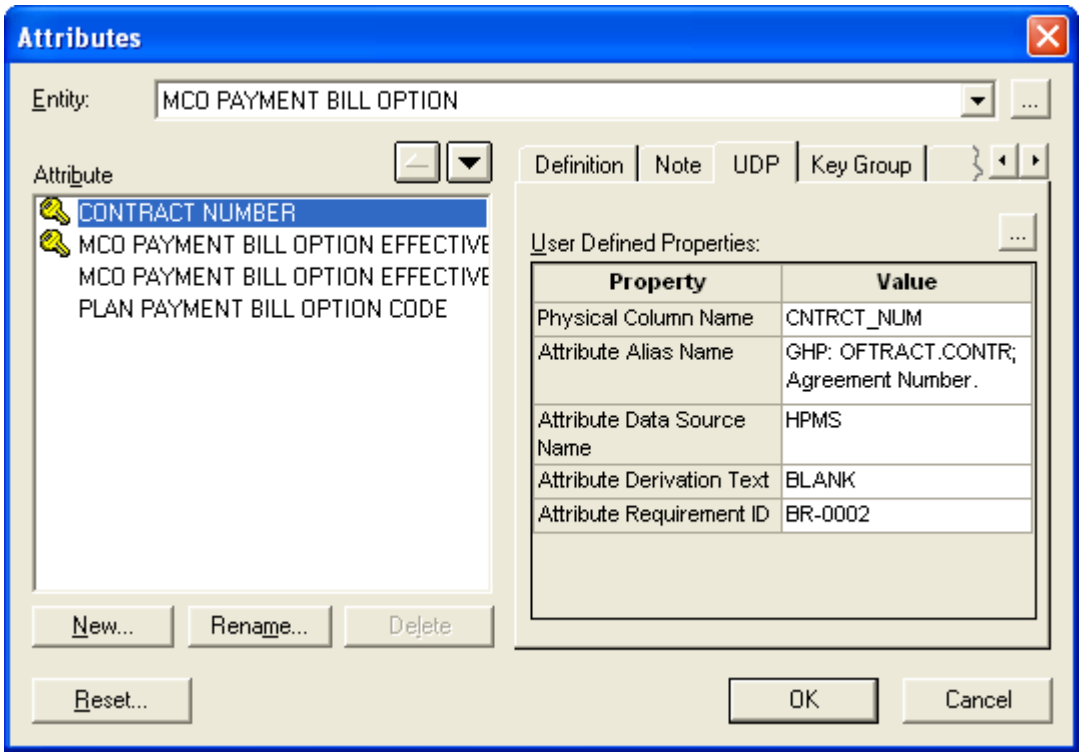


Exhibit 15. Attribute UDPs for a Project Logical Data Model



1.4. Data Modeling tool standard for Creating Project Physical Data Models

Introduction
All-Fusion ERwin Data Modeler is the standard data-modeling tool at CMS. Use of any other software tool for the purpose of developing Physical Data Models is prohibited.

Responsibilities
All entries in the data modeling tool to create the *first cut* Project Physical Data Model are performed by the *Project Data Analyst.*

Data Model Template File
A Model Template File and a Naming Standard File are available from the DA Standard Tools page, which is accessible from the main Data Administration web page. The template file contains the CMS standard User Defined Properties (UDPs). The Naming Standards File enables automatic naming features in the data modeling tool.

Refer to the *HELP* in the data modeling tool for information on how to implement and use the naming file.

The use of this model template is required for *Project Physical Data Models* being developed on or after October 1, 2005, which supports all new development projects. Models that will be used to modify databases that existed before October 1, 2005 are required to preserve any UDPs already defined in those previous models. Database enhancement projects are encouraged to incorporate and populate the new standard UDPs to the greatest practical extent. The new UDPs themselves can be added by using the ERwin compare utility to update the project PDM. (Refer to DM G-022 Guideline for Using ERwin Complete Compare to Import Standard Physical Model UDPs.) Then they can be populated using the UDP tab of the regular table, column and model property dialogues.

When you open either template up in ERwin all you will see is a blank screen. These templates contain absolutely no entitles, tables, or any other diagram objects. The templates exist solely to offer a set of standard UDPs that can be used in any data model.

The ERwin model properties for a Project Physical Data Model are to be specified according to the table which follows.

Erwin Model Properties for a Project Physical Data Model

Model Property	**Format / Description**	**Reqd**	**UDP**
Model Definition	Provide a brief description of the business project whose high-level data requirements are represented by the entities and relationships to be diagramed in the Conceptual Data Model. The model definition describes the purpose and status of the model in a few sentences of text. Example**: The DB2 physical data model for the XYZ database.** (Graphic) See: DM OP- 028 Operating Procedure for Naming and Defining Data Models.	●	
Model History Options	Select all of these options. (Graphic)	●	
Model Name	A Physical Data Model is to be named in the following manner: *system acronym + ("relational"/"dimensional") + model type (EDM/CDM/LDM/PDM) + approval date (or the save date for models in development) in yyyymmdd format*. Example: **MBD relational PDM 20040726**. (Graphic) See: DM OP- 028 Operating Procedure for Naming and Defining Data Models.	●	
Model Notation	Select IDEF1X. (Graphic)	●	
Model Type	Designate the model as a pure logical or pure physical model, according to the kind of objects contained in the model. For a Physical Data Model, the type must be physical. (Graphic)	●	
PDM Create Date	Enter the date designated by Central DA for the first PDM generated/ developed. (This date will determine the standards that apply to the Model.) (Graphic)	●	●
PDM DA Signoff Date	Enter the date when the PDM received its baseline/ final Central DA sign-off. (Graphic)	●	●
PDM Last Change Description	A brief narrative summarizing the nature of the changes resulting in the current model version. (Graphic)	●	●
PDM System Contact Name	Enter the name of the organization and person who is responsible for approving the physical database design Model. (Graphic)	●	●

Table Properties

The ERwin properties for each table in a Physical Data Model are to be specified according to the following description.

Table Property	Format / Description	Reqd	UDP
Logical Entity Name	The name of the corresponding Entity defined in the corresponding LDM for this PDM. Not required for "physical only" tables. (Graphic)	●	●
Table Comment	The narrative explanation of the meaning of a row in the Table. Example: **MBD_CONTRACT - A binding agreement between CMS and an Organization that enables eligible Medicare beneficiaries to obtain medical services from the organization in exchange for monthly payments. Both current and historical information is retained.** (Graphic)	●	
Table Name	The user assigned symbolic identifier of the Table. Type *Table Names* in uppercase with words separated by an underscore character throughout the model. (Graphic)	●	
Table Physical Only Switch	The indication of whether or not the Table has a corresponding Entity in an LDM. (Graphic)	●	
Table Requirement ID	A reference to the requirement(s) that justify the existence of the Entity in the model. The format of the reference to the DOORS Tracking ID is BR-#### for business requirements, SR-#### for system requirements and CR-#### for a change request. Multiple requirements are separated by semicolons. (Graphic)		●

Column Properties

The ERwin properties for each data attribute in a Physical Data Model are to be specified according to the table which follows.

Column Property	Format / Description	Reqd	UDP
Column Comment	The narrative explanation of the meaning of an instance of the column. (Graphic) Example: **MCO_NAME - Legal entity name of the organization.**	●	
Column Data Source Name	The CMS database, external data feed, manual data entry process or software process from which the attribute takes its value. The format for CMS sources is CMS.database.table.column. The format for external sources is org.specification.pubdate.record.recordtype.field. Recordtype is used where there are multiple record formats and may be omitted if there is only one format for the record specification. The format for software processes is application.programname or packagename.servicename. The format for manual data entry processes is application.formname.fieldname. If this level of detail is not available, indicate the source in terms of a user role, type of stakeholder, organization, business process, system, program, database, file or data exchange standard. (Graphic)	●	●
Column Derivation Text	The narrative explanation of the logic used to transform information from one or more other columns or fields into the value for this Column. (Graphic)		●
Column Domain Name	The name of the domain which defines the datatype, default value and valid values of the column. (Graphic)	●	
Column Name	The user assigned symbolic identifier of the Table. Type *Column Names* in uppercase with words separated by an underscore character throughout the model. (Graphic) If the column is serving as a foreign key, assign a role name where necessary.	●	
Column Null Option	An indication of whether or not the Column must assume a non-null value when a row is created. (Graphic)	●	
Column Overriding Datatype Name	The name of the datatype which overrides the one specified in the Domain which governs the Column. (Graphic) (Required if no Domain is specified.)		
Column Overriding Default Value Name	The name of the Default Value which overrides the one specified in the Domain which governs the Attribute. (Graphic)		
Column Overriding Validation Rule Name	The name of the Validation Rule which overrides the one specified in the Domain which governs the Attribute. (Graphic)		

Column Property	Format / Description	Reqd	UDP
Column Physical Only Switch	The indication of whether or not the Column has a corresponding Attribute in an LDM. (Graphic)	●	
Column Requirement ID	A reference to the requirement(s) that justify the existence of the Column in the model. The format of the reference is BR-#### for business requirements, SR-#### for system requirements and CR-#### for a change request. Multiple requirements are separated by semicolons. (Graphic)	●	●
Logical Attribute Name	The name of the corresponding Attribute, specified in the LDM, upon which this Column's design was based. The format of this reference is ModelName.VersionDate.EntityName.AttributeName. Not required for "physical only" columns. (Graphic)	●	●
Primary Key Switch	An indication of whether or not the Column is part of the Table's primary key. (Graphic)	●	

Data Relationships

Verb phrase in all lower case
Relationship type
Specify identifying or non-identifying relationship.
Physical Only Indicator: Specify *Physical Only* status, if applicable.
Specify cardinality.

ERwin Screen Snapshots for Creating Physical Data Models

Exhibit 16. Model Properties for a Project Physical Data Model

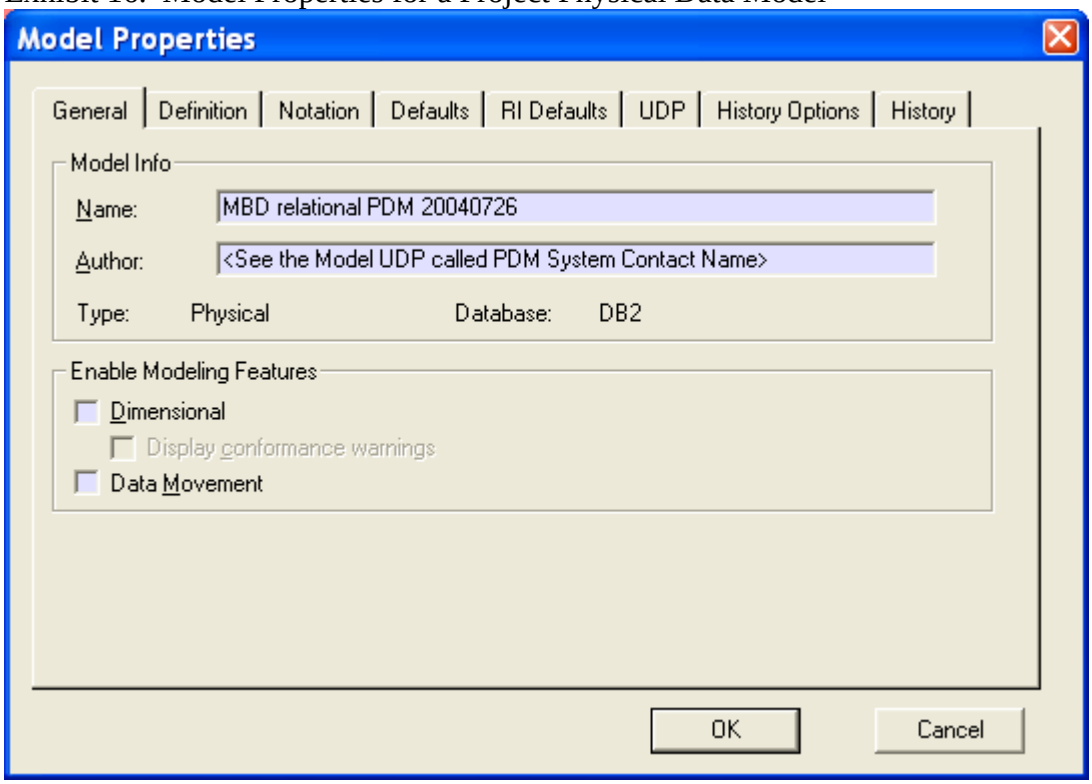


Exhibit 17. Example Model definition for a Project Physical Data Model

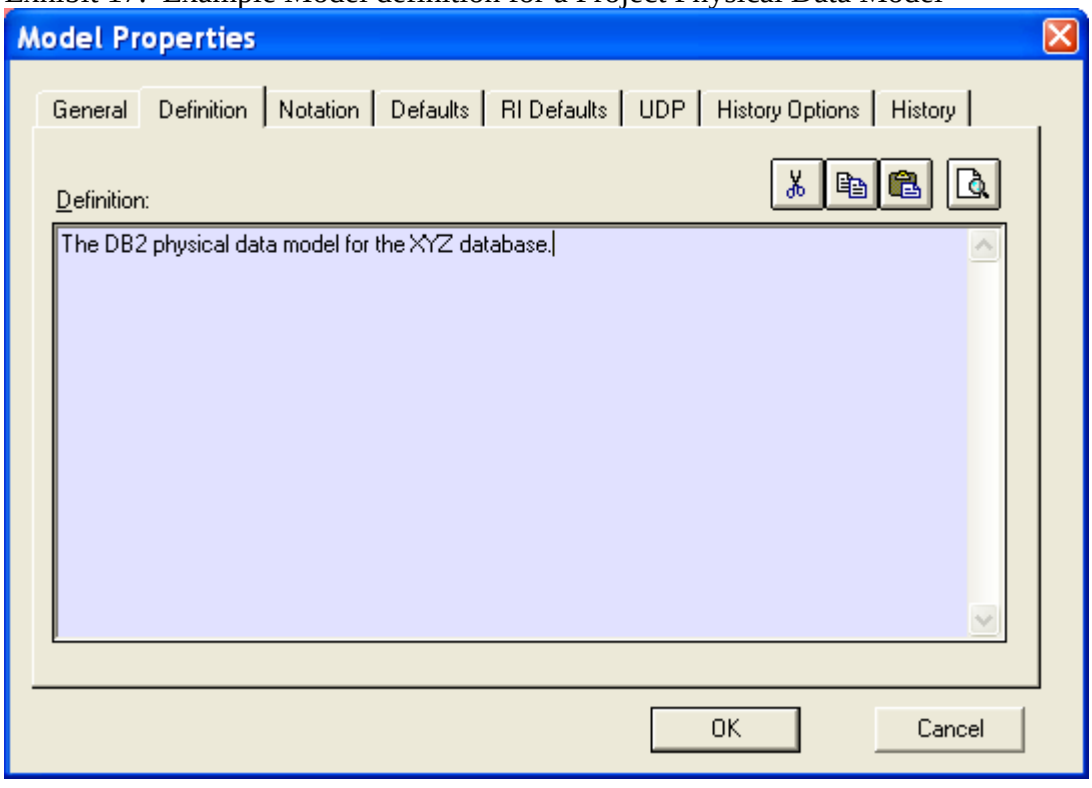


Exhibit 18. Example Model Type for a Project Physical Data Model

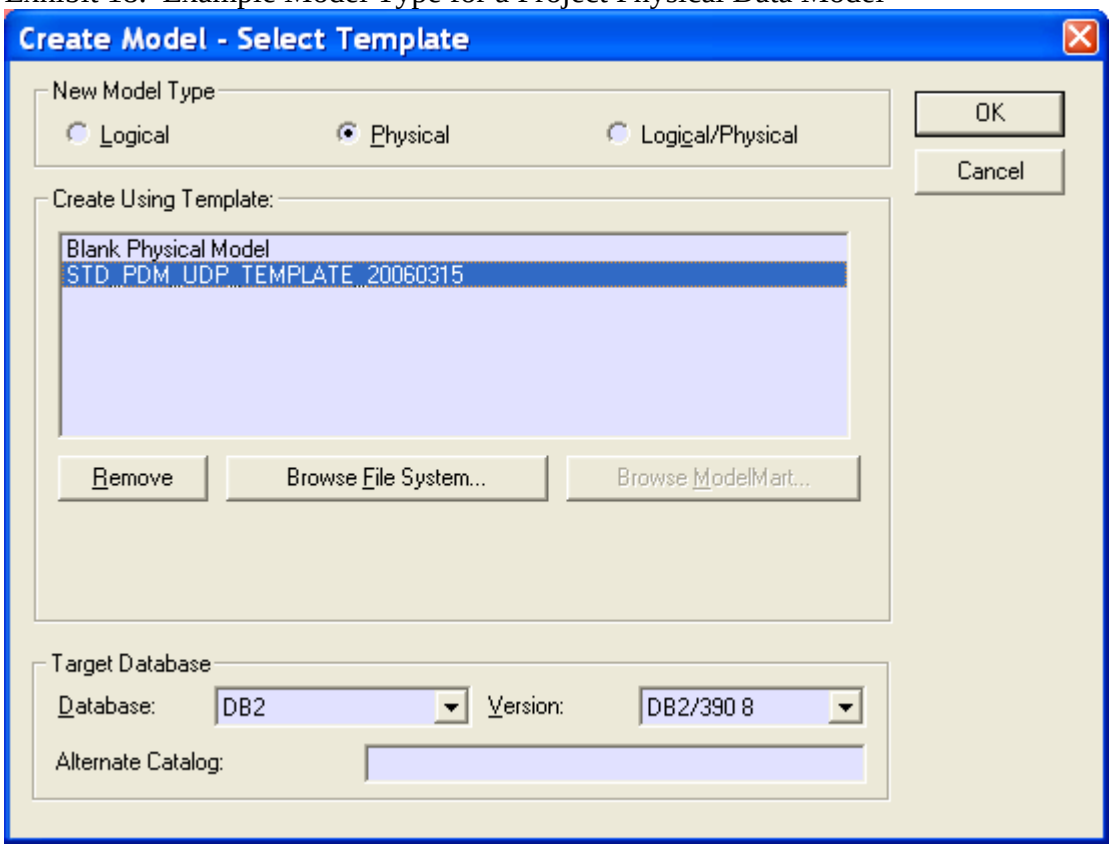


Exhibit 19. Example Model Notation Option for a Project Physical Data Model

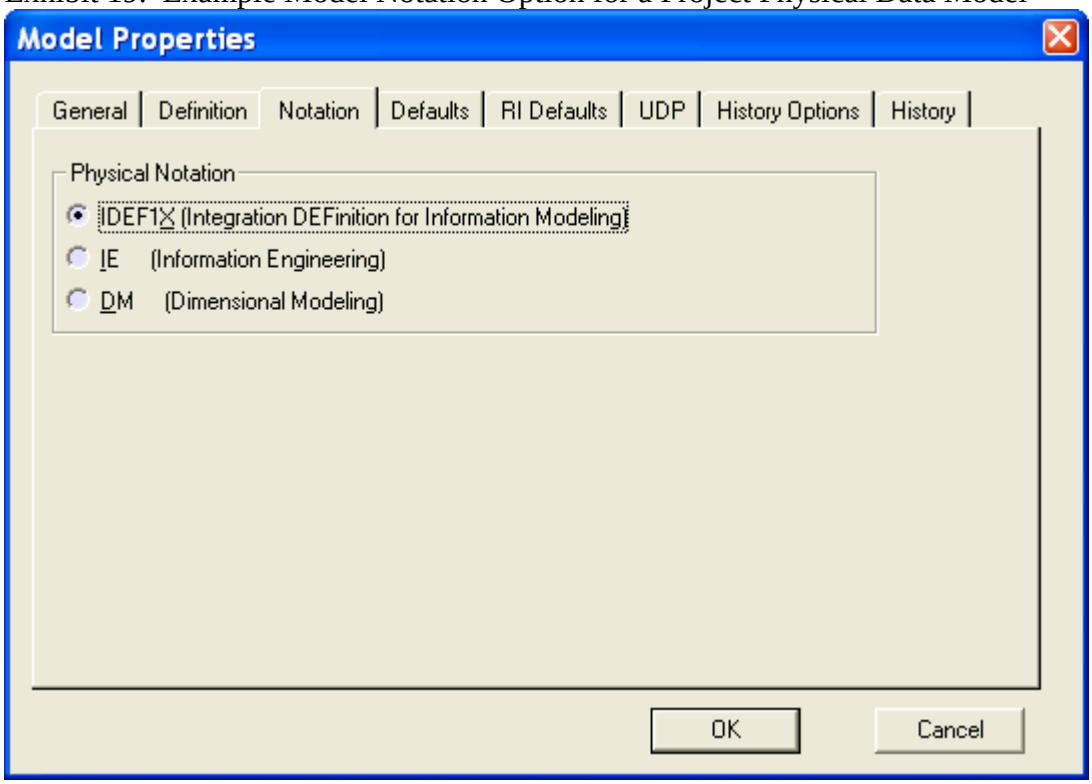


Exhibit 20. Model UDPs for a Project Physical Data Model

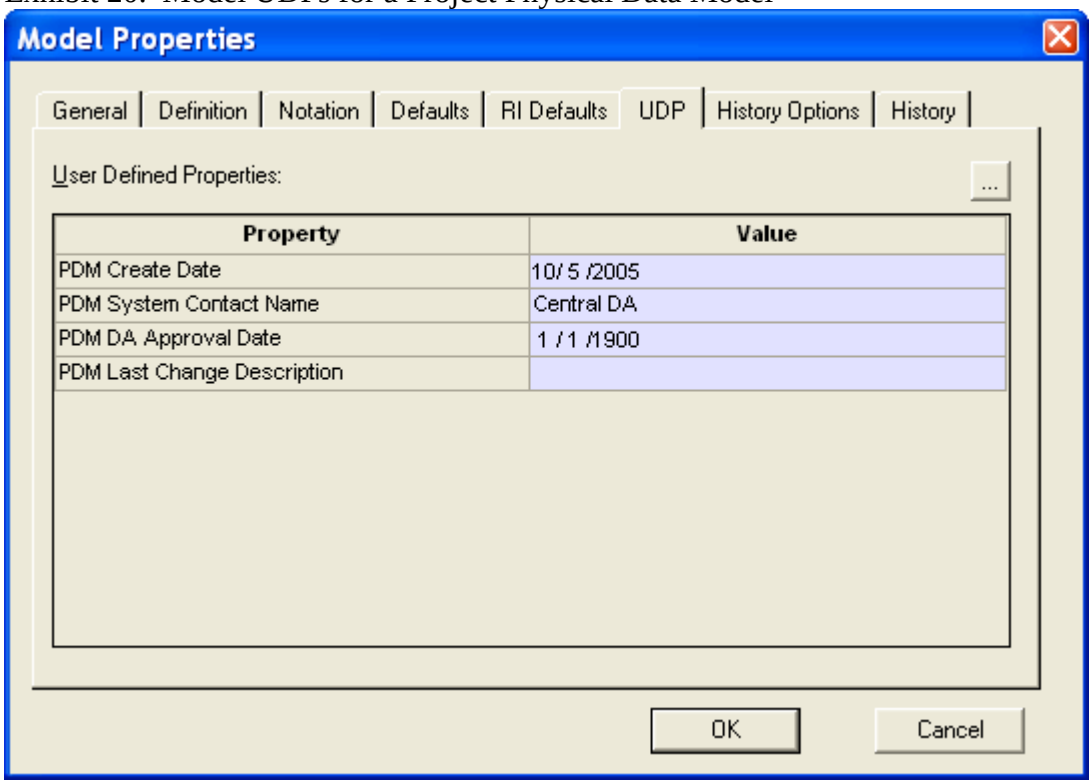


Exhibit 21. Example Data Structure Diagram for a Project Physical Data Model

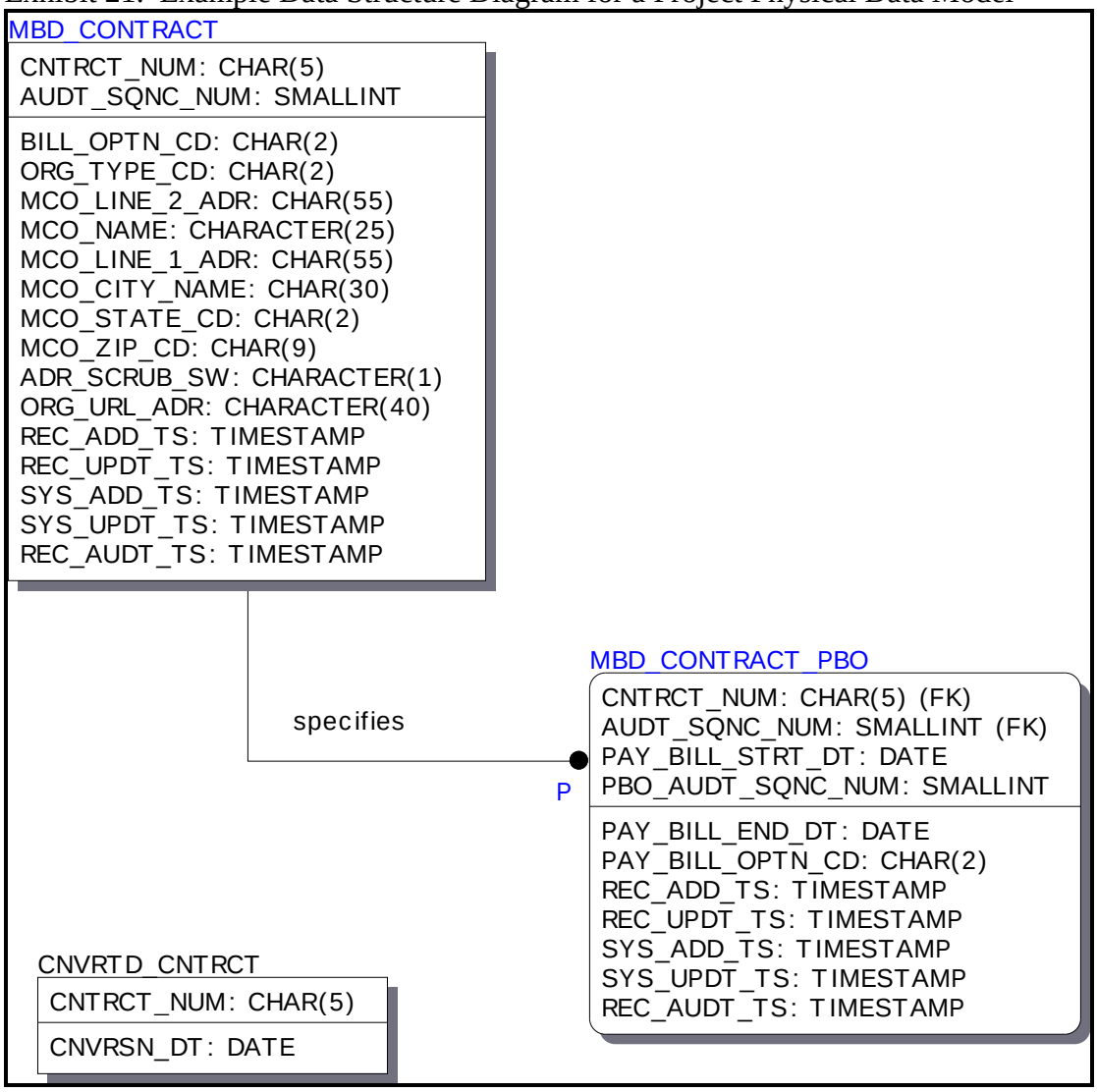


Exhibit 22. Example Table Definition for a Project Physical Data Model

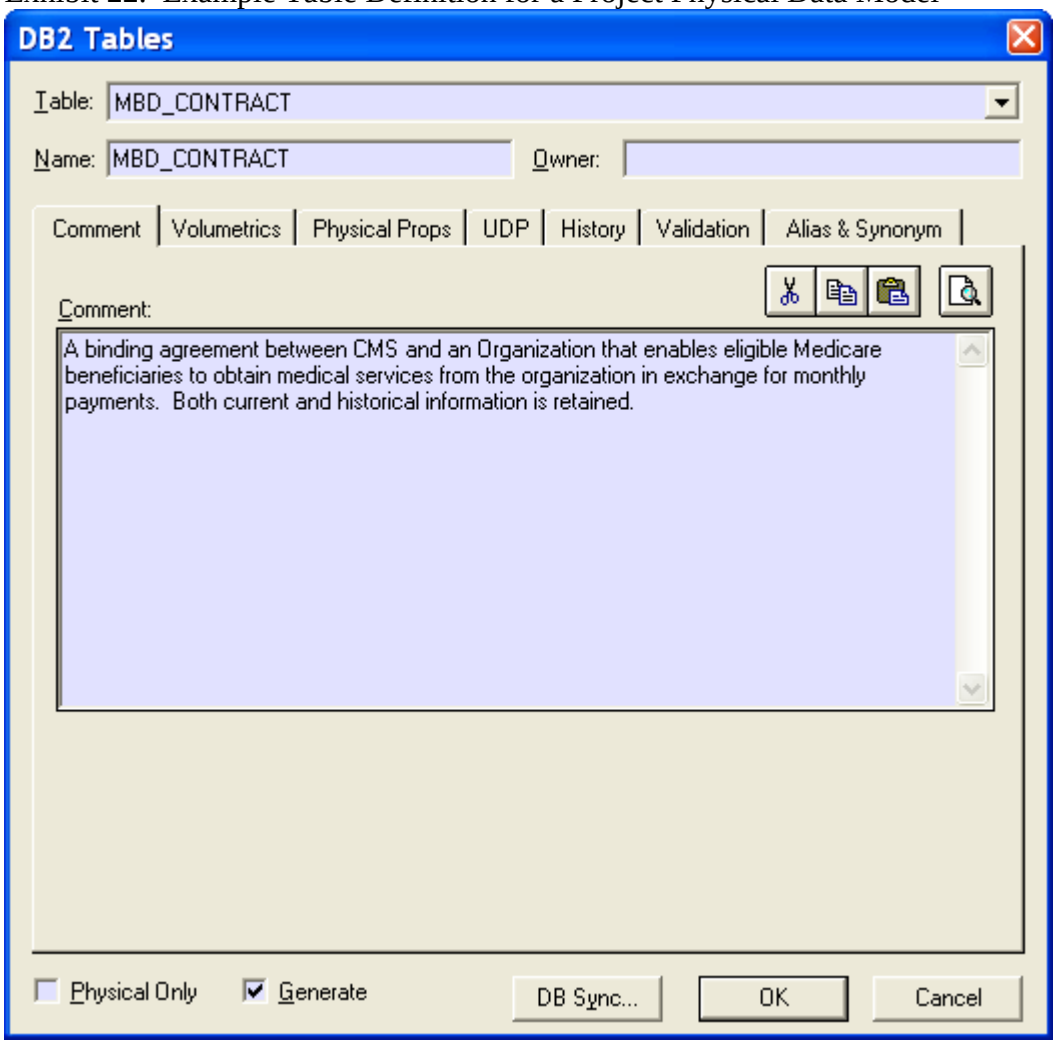


Exhibit 23. Example Table UDPs for a Project Physical Data Model

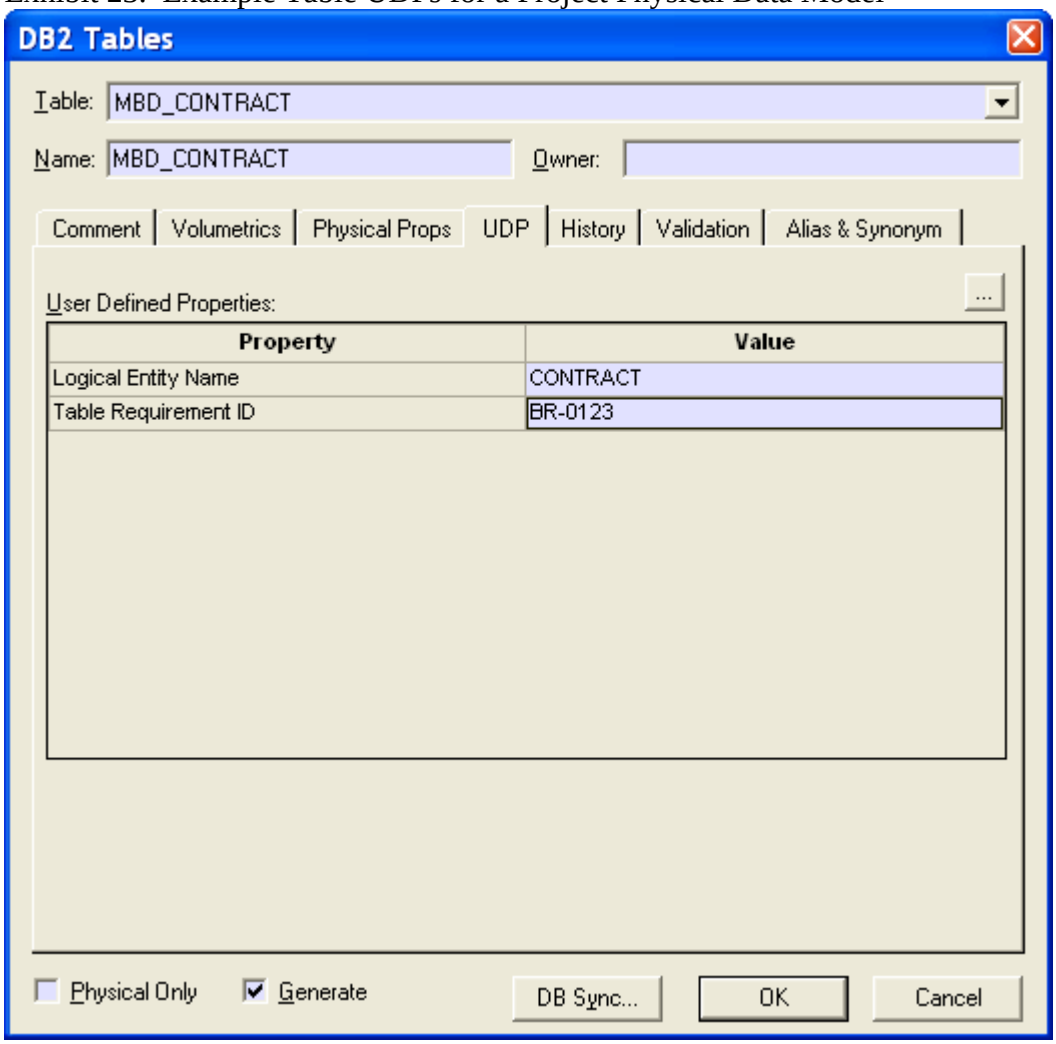


Exhibit 24. Example Column Comment for a Project Physical Data Model

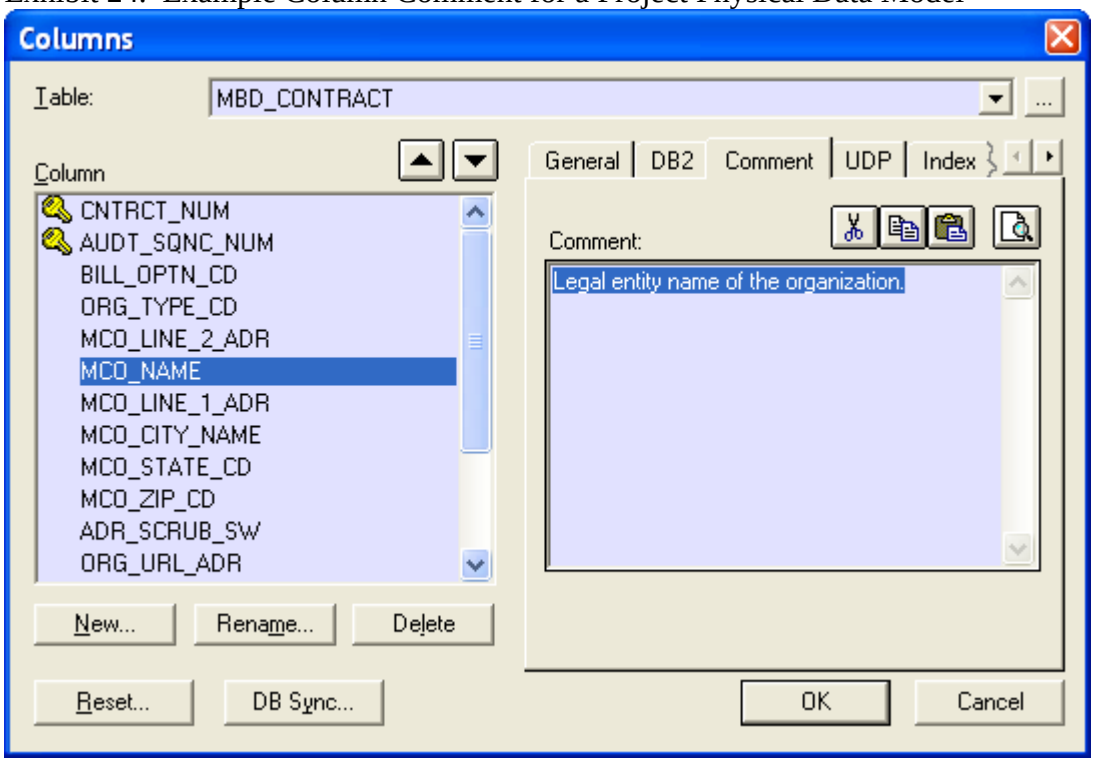


Exhibit 25. Column General Properties for a Project Physical Data Model

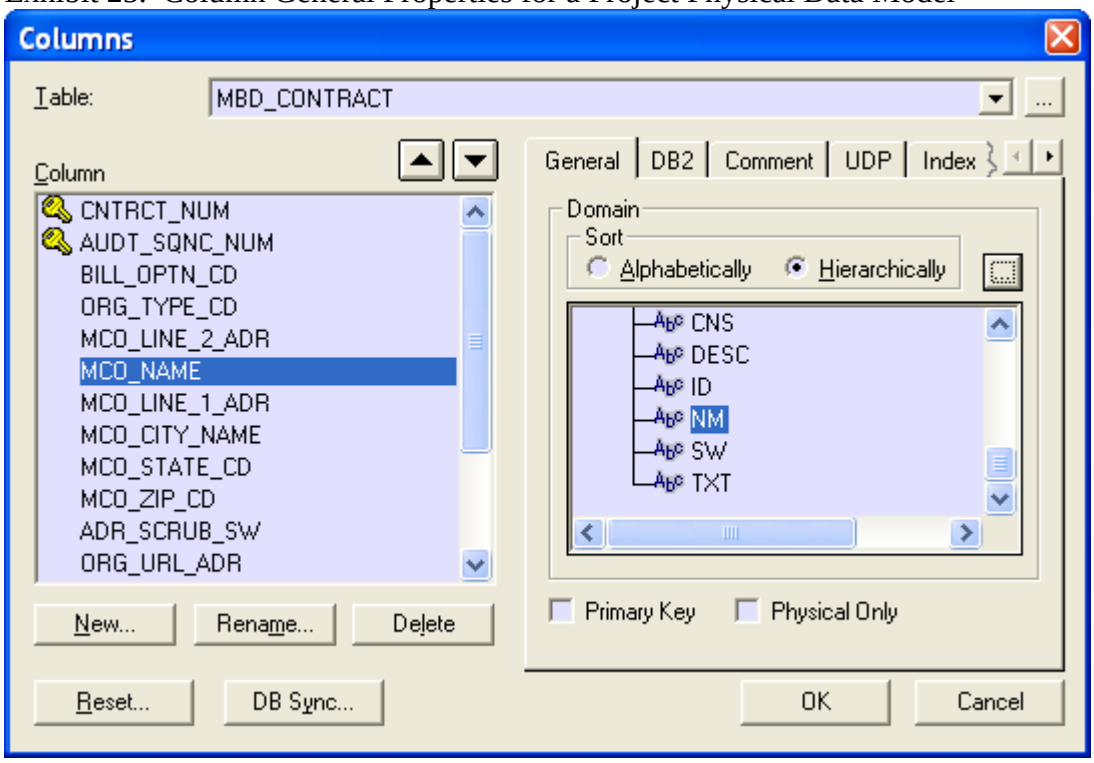


Exhibit 26. Example Database Specific Properties for a Project Physical Data Model

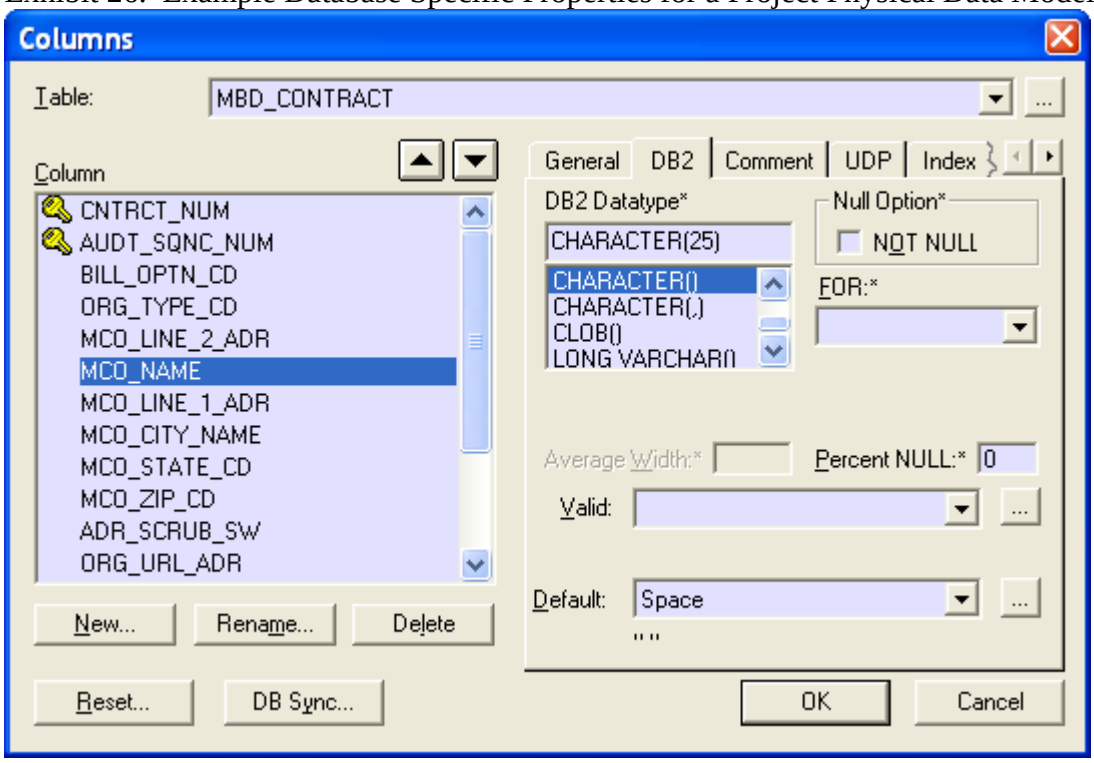


Exhibit 27. Example Column UDPs for a Project Physical Data Model